Center For Human-Machine Systems Research

Intent Inferencing with a Model-Based Operator's Associate

Patricia M. Jones
Christine M. Mitchell
Kenneth S. Rubin

Report No. 88-2

August 1988

TECH

School of Industrial and Systems Engineering
Georgia Institute of Technology
A Unit of the University System of Georgia
Atlanta, Georgia 30332

INTENT INFERENCING WITH A MODEL-BASED OPERATOR'S ASSOCIATE

Patricia M. Jones, Christine M. Mitchell, and Kenneth S. Rubin*

Center for Human-Machine Systems Research

School of Industrial and Systems Engineering

Georgia Institute of Technology

Atlanta, GA 30332

* Currently with ParcPlace Systems, 2400 Geng Rd., Palo Alto CA 94303

ABSTRACT

This paper describes a portion of the OFMspert (Operator Function Model Expert System) research project. OFMspert is an architecture for an intelligent operator's associate or assistant that can aid the human operator of a complex, dynamic system. Intelligent aiding requires both understanding and control. This paper focuses on the understanding (i.e., intent inferencing) ability of the operator's associate. Understanding or intent inferencing requires a model of the human operator; the usefulness of an intelligent aid depends directly on the fidelity and completeness of its underlying model. The model chosen for this research is the operator function model (OFM) (Mitchell, 1987). The OFM represents operator functions, subfunctions, tasks, and actions as a heterarchic-hierarchic network of finite state automata, where the arcs in the network are system triggering events. The OFM provides the structure for intent inferencing in that operator functions and subfunctions correspond to likely operator goals and plans. A blackboard system similar to that of HASP (Nii et al., 1982) is proposed as the implementation of intent inferencing function. This system postulates operator intentions based on current system state and attempts to interpret observed operator actions in light of these hypothesized intentions. The OFMspert system built for this research is tailored for the GT-MSOCC (Georgia Tech Multisatellite Operations Control Center) simulation. The GT-MSOCC OFMspert has been the subject of rigorous validation studies (Jones, 1988) that demonstrate its validity as an intent inferencer.

INTRODUCTION

Computational representations and models have been constructed for "understanding" human behavior in many applications; e.g., understanding natural language (Winograd, 1972) and understanding stories (Schank and Abelson, 1977). Artificial intelligence has developed many representational formalisms and control strategies that are intended to mimic "intelligent" behavior (cf Cohen and Feigenbaum, 1982). In the field of human-machine systems research, AI techniques offer powerful methodologies for understanding human behavior in the context of human-machine interaction.

Our particular concern is with human-machine interaction in the control of complex dynamic systems (e.g., nuclear power plants). Such systems are highly automated; thus, the human operator acts as a supervisory controller (Sheridan and Johannsen, 1976; Rasmussen, 1986; Wickens, 1984). Supervisory control typically consists of routine monitoring and fine-tuning of system parameters. However, in the event of abnormal or emergency situations, the human operator is expected to detect, diagnose, and compensate for system failures. The ability of a supervisory controller to cope with such situations can be severely limited. Wickens (1984) cites several problems with supervisory control: an increased monitoring load; a "false sense of security" whereby the operator trusts the automation to such an extent that any human intervention or checking seems unnecessary; and "out-of-the-loop familiarity" that implies a reduced ability to cope with non-routine situations.

An important question then becomes how to improve system performance and safety in supervisory control. The answer is not to automate the human out of the system; today's technology cannot match the human's ability to cope with uncertain and novel situations (Chambers and Nagel, 1985). Rather, automated systems must support the human operator. Given that the human will remain an integral part of a complex system, a potential approach to advanced automation is that of "amplifying" rather than automating human skills (Woods, 1986).

The OFMspert (Operator Function Model Expert System) project is an effort to develop a theory of human-computer interaction in supervisory control. OFMspert itself is a generic architecture for a computer-based operator's associate. The operator's associate (and similarly, the Pilot's Associate (Rouse et al. 1987; Chambers and Nagel, 1985)) represents a design philosophy that allows the human to remain in control of a complex system. The computer-based associate is a subordinate to which the human operator can delegate control activities. The associate also actively monitors system state and operator actions in order to timely, context-sensitive advice, reminders, and suggestions. The intent is to provide intelligent support for the human operator.

The intelligence and utility of the operator's associate rest on its abilities to understand the operator's current intentions in order to provide context-sensitive advice and assume responsibility given for portions of the control task. Models of human-machine interaction offer a variety of frameworks for understanding human behavior (i.e., inferring intentions) in the control of a complex dynamic system (see Jones and Mitchell, 1987, and Jones, 1988, for a review). Knowledge-based problem solving strategies are tools for implementing and reasoning with the knowledge represented in the human-machine interaction model. OFMspert combines a particular human-machine interaction model (the operator function model (OFM) (Mitchell, 1987)) and knowledge-based problem solving approach (the blackboard model of problem solving (Nii, 1986)) to provide the understanding capability necessary for an effective operator's associate (Rubin, et al., 1987). In the next sections, the OFM and the blackboard model of problem solving are described. Next, ACTIN (Actions Interpreter), the intent inferencing component of OFMspert, is discussed, along with a detailed example of how ACTIN infers operator intentions dynamically. Finally, experimental results that validate ACTIN's intent inferencing ability are considered.

THE OPERATOR FUNCTION MODEL

The operator function model (OFM) (Mitchell, 1987) provides a flexible framework for representing operator functions in the control of a complex dynamic system. The OFM represents how an operator might organize and coordinate system control functions. Mathematically, the OFM is a hierarchic-heterarchic network of finite-state automata. Network nodes represent operator activities as operator functions, subfunctions, tasks, and actions. Operator functions are organized hierarchically as subfunctions, tasks, and actions. Each level in the network may be a heterarchy, i.e., a collection of activities that may be

ORIGINAL PAGE IS
OF POOR QUALITY

performed concurrently. Network arcs represent system triggering events or the results of operator actions that initiate or terminate operator activities. In this way, the OFM accounts for coordination of multiple activities and dynamic focus of attention.

Historically, the OFM is related to the discrete control modeling methodology (Miller, 1985; Mitchell and Miller, 1986). The OFM is distinguished by its modeling of both manual and cognitive operator actions in the context of system triggering events. Manual actions are system reconfiguration commands. Cognitive actions include information gathering and decision making that are typically supported by information requests.

The OFM is a prescriptive model of human performance in supervisory control. Given system triggering events, it defines the functions, subfunctions, tasks, and actions on which the operator should focus. Used predictively, the OFM generates expectations of likely operator actions in the context of current system state. Used inferentially, the OFM defines likely operator functions, subfunctions, and tasks that can be inferred based on operator actions and system state. Thus, the OFM for a particular domain defines the knowledge needed to perform intent inferencing. What is needed next is a problem solving strategy to use this knowledge.

THE BLACKBOARD MODEL OF PROBLEM SOLVING

OFMspert's intent inferencing component, called ACTIN (Actions Interpreter), uses the HASP blackboard model of problem solving (Nii et al, 1982; Nii, 1986). The HASP blackboard is one of the few artificial intelligence systems that explicitly addresses real-time problem solving in dynamic environments.

The blackboard model of problem solving consists of three components: the blackboard, knowledge sources, and blackboard control. The blackboard is a data structure on which the current best hypothesis of the solution is maintained and modified. The hypothesis is represented hierarchically, at various levels of abstraction, and evolves incrementally over time as new data become available or old data become obsolete. Domain-specific knowledge is organized as a collection of independent knowledge sources. Knowledge sources are responsible for posting and interpreting information on the blackboard. Blackboard control applies knowledge sources opportunistically; that is, in either a top-down or bottom-up manner, depending on what is more appropriate in the current context.

The blackboard model of problem solving is compatible with the knowledge represented in the OFM. Both models use a hierarchical representation. The blackboard knowledge sources provide a modularity that naturally represents much of the domain knowledge contained in the OFM arcs. The opportunistic control strategy offers the dynamic flexibility necessary for inferring intentions in real time. ACTIN combines the OFM representation of domain knowledge and the blackboard model of problem solving to dynamically construct and assess current operator intentions.

ACTIONS INTERPRETER (ACTIN)

ACTIN's blackboard represents operator intentions as a hierarchy of goals, plans, tasks, and actions that correspond to the OFM's hierarchy of functions, subfunctions, tasks, and actions. Goals are currently instantiated functions, plans are currently instantiated subfunctions, and so on. In some respects, ACTIN is a process model that uses the blackboard problem solving method to build a dynamic representation of current operator intentions based on the OFM's static knowledge (Wenger, 1987).

The general mechanism for the blackboard approach to intent inferencing is as follows. Given an OFM, currently hypothesized goals, plans, and tasks (GPTs) or sometimes additional plans and tasks (PTs) for an existing goal are placed on the blackboard in response to system triggering events. The blackboard incorporates operator actions into the representation with opportunistic reasoning. Thus, actions can be immediately interpreted as supporting one or more current goals, plans, and tasks; and goals, plans, and tasks can be inferred on the basis of operator actions.

Construction knowledge sources are responsible for building the representation of goals, plans, tasks, and actions. These knowledge sources can further be characterized as either model-driven or data-driven. Model-driven knowledge sources are those that post GPT information on the blackboard in response to system triggering events as defined by the OFM. Data-driven knowledge sources are those that post operator actions and attempt to infer support for any current tasks on the blackboard. Data-driven knowledge sources may also postulate GPT information on the basis of operator actions. Assessment knowledge sources are responsible for evaluating the extent to which operator actions support currently hypothesized goals, plans, and tasks. Assessments are always made in the context of a particular goal or plan which forms the context for possible advice or reminders.

In order to illustrate ACTIN's dynamic intent inferencing, it is first necessary to describe the application domain for which our OFMspert was built: the Georgia Tech Multisatellite Operations Control Center (GT-MSOCC). After describing GT-MSOCC and its OFM, an example of ACTIN's intent inferencing is presented.

GT-MSOCC: APPLICATION DOMAIN

GT-MSOCC is a real time, interactive simulation of MSOCC, a NASA ground control station for near-earth satellites (Mitchell, 1987). MSOCC is a facility for capturing and processing data sent by satellites (see Figure 1). GT-MSOCC is a research domain designed to support theoretical and empirical research on human-computer interaction in the context of a complex dynamic system. It is a high fidelity simulation of the operator interface to an actual NASA ground control system. For more detail, see Mitchell, 1987.

GT-MSOCC operator activities are defined by the GT-MSOCC OFM. At the highest level of the GT-MSOCC operator function model are major operator functions and the system events that cause the operator to transition among functions (see Figure 2). This level of description represents operator goals in the context of current system state. The arcs define system events that trigger a refocus of attention or the addition of a function to the current set of operator duties.

The default high-level function is to control current missions. This involves the subfunctions of monitoring data transmission and hardware status, detection of data transmission problems, and compensation for failed or degraded equipment. Each subfunction is further defined by a collection of tasks, which in turn are supported by operator actions (system reconfiguration commands or display requests).

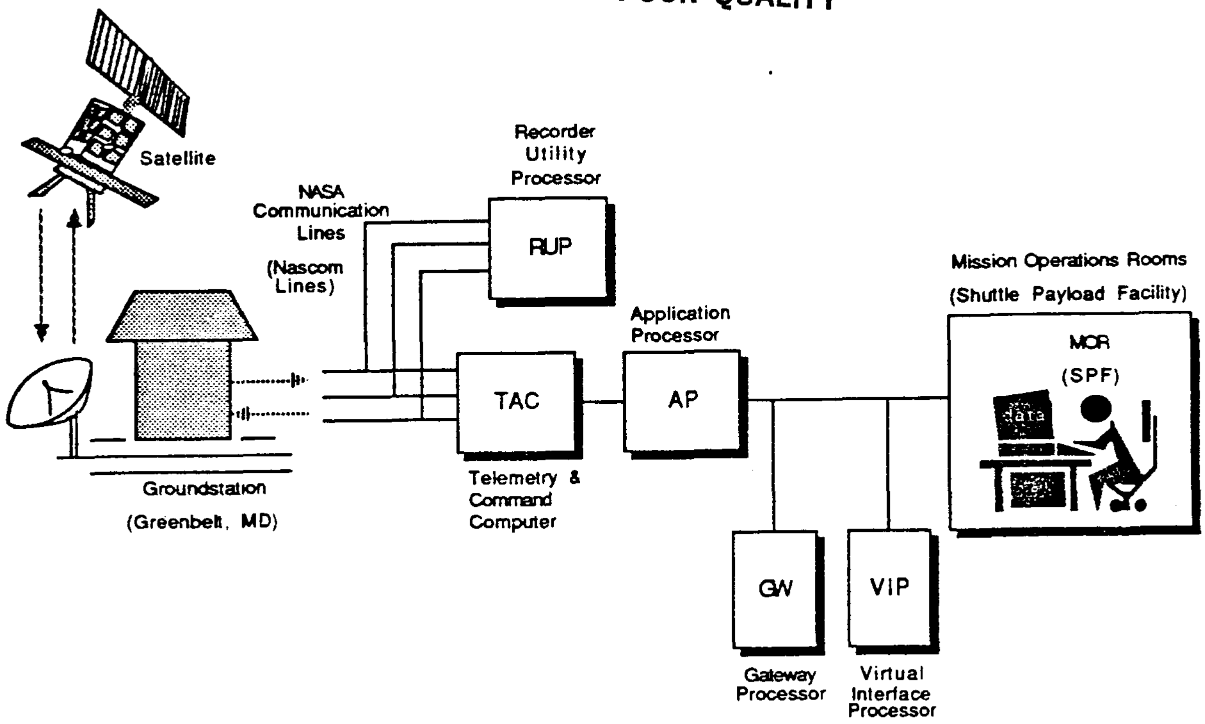
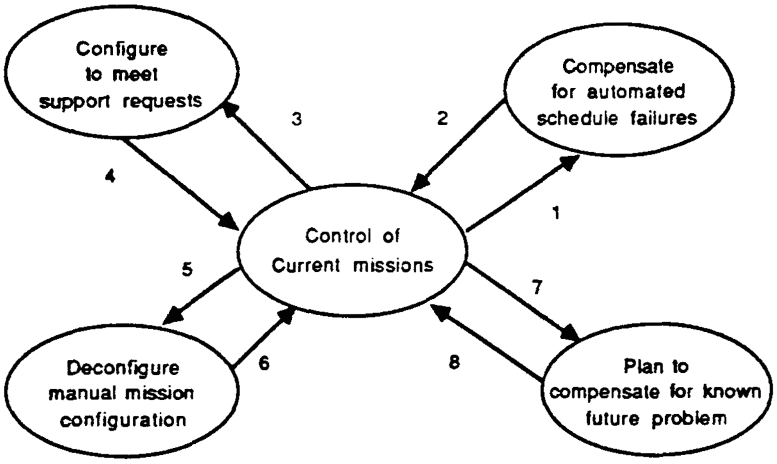


Figure 1. Multisatellite Operations Control Center (MSOCC)

System triggering events cause the operator to focus attention on other high-level functions. An unscheduled support request causes the operator to shift to the "configure to meet support requests" function. An error message from the automatic scheduler causes the operator to transition to the function to compensate for the automated schedule failure. A request to deconfigure a mission causes the operator to shift to the function of deconfiguring a manual mission configuration. Finally, the operator may engage in long-term planning in the absence of other system triggering events. Upon the termination of these other functions, the operator resumes the default control of current missions function. Functions may be terminated by their successful completion or the determination that they cannot be completed.

ACTIN'S INTENT INFERENCING WITH GT-MSOCC

In this section, a detailed example of ACTIN's intent inferencing is provided in the context of GT-MSOCC. Table 1 shows the organization of GT-MSOCC goals, plans, tasks, and actions, as defined by the GT-MSOCC OFM. Given system triggering events, ACTIN's model-driven knowledge sources post the


Configure to meet support requests
Compensate for automated schedule failures
Control of Current missions
Deconfigure manual mission configuration
Plan to compensate for known future problem
1
2
3
4
5
6
7
8

1. Error message received from the automatic scheduler.
2. Compensation completed or unable to compensate.
3. Unscheduled support request received by the operator.
4. Request configured or unable to meet the request.
5. Message received that a manually configured mission is completed.
6. Deconfiguration completed.
7. Operator summons schedule and/or mission template pages when no other triggering event takes place.
8. Terminate planning function.

Figure 2. GT-MSOCC Operator Functions

appropriate goal, plan, and task (GPT) structures on the blackboard. When operator actions occur, ACTIN's data-driven knowledge sources post actions on the blackboard and attempt to "connect" the actions to tasks which they support. This "connection" between actions and tasks defines ACTIN's intent inferencing capability. The knowledge of appropriate inferences of intent is contained in a data structure that matches actions to task types. Data-driven knowledge sources consult this structure to determine that task type(s) that a current operator action can support. They then search the blackboard's task level of abstraction for those types, and connect the action to all appropriate tasks.

To illustrate ACTIN's dynamic construction of operator intentions, consider the following scenario from GT-MSOCC. The scenario is described in terms of GT-MSOCC system events and operator actions, which then cause activity on the blackboard. ACTIN's intent inferencing results in statements written to a logfile. In the accompanying figures, the current blackboard structure is shown, along with ACTIN's inferences of intent.

1). The PM mission is automatically configured. ACTIN's model-driven knowledge sources post the goal to control the current mission (CCM) for PM. This goal is comprised of two plans: to monitor data

ORIGINAL PAGE IS
OF POOR QUALITY

Table 1. GT-MSOCC Goals, Plans, Tasks, and Actions

Goals	Plans	Tasks	Actions
Control current mission CCM	Monitor software (MSW)	Check MOR (CMOR)	telem
		Check endpoints (CEND)	rup/gw/vip/cms telem
	Monitor hardware (MHW)	Check hardware (CHW)	–
Manual Configure Request CON	Check system constraints (CSC)	Check current number of missions (CCNM)	–
		Check mission schedule (CMS)	msocc sched, msn scheds
		Check scheduled number of missions (CSNM)	msocc sched,pending
	Check mission requirements (CMR)	Check mission template (CMT)	msocc sched. msn scheds
	Identify candidate hardware (ICH)	Find current (FCUR)	–
		Find unscheduled (FUSC)	equip scheds, avails
	Answer question (ANQ)	Execute answer(XAN)	operator answer
		Execute configure (XCON)	manual config. (MCON) events
Compensate for Schedule Failure CFSF	Reconfigure (RCON)	Find duration (FDUR)	telem, pending
		Execute reconfigure(XRCO)	man. reconfig (MRCO), events
		For each equipment:	
		Find current (FCUR)	–
		Find unscheduled (FUSC)	equip scheds, avails
——	Manual Deconfigure Request (DCON)	Execute deconfigure (XDCO)	man. deconfig(MDCO), events
——	Troubleshoot (TBLS)	Check endpoints (CEND)	gw/rup/cms/vip telem
		Check interior (CIN)	nas/tac/ap/modlan telem
——	Replace(HRPL or SRPL)	Find duration (FDUR)	telem, pending
		Find current (FCUR)	–
		Find unscheduled (FUSC)	equip scheds, avails
		Execute replace(XRPL)	replace (RPL)

transmission or software (MSW) and to monitor hardware status (MHW). Each plan is composed of one or more tasks. The monitor software plan consists of two tasks: to check data flow at the MOR (CMOR) and to check data flow at endpoint equipment (CEND). The monitor hardware plan consists of the single task to check hardware status (CHW). This entire GPT structure defines the control of current mission function prescribed by the OFM. When PM is configured, ACTIN's knowledge sources retrieve the control of current mission GPT structure, fill in mission-specific information (e.g., the name of this particular mission is PM), and post the structure on the blackboard. The resulting blackboard is shown in Figure 3a.

2). Another mission (Geographic Explorer, or GEO) is configured. In the same way the control of current mission GPT was posted for PM, a control of current mission GPT for GEO is also posted. The resulting blackboard is shown in Figure 3b.

3). The operator requests the main telemetry page ("telem"). ACTIN's data-driven knowledge sources determine that the current action type is "telem" and that actions of this type potentially support the tasks of checking the MOR (CMOR) and finding the duration (FDUR) of current missions. Upon examining the tasks level of the blackboard, the knowledge sources find that two eligible tasks are posted: the CMOR tasks for PM and GEO. Thus, the "telem" action is posted and connected to the CMOR tasks. The resulting blackboard is shown in Figure 3c.

4). The operator requests the gateway telemetry page ("GwTelem"). ACTIN's data-driven knowledge sources determine that the current action type is "GwTelem" and that actions of this type potentially

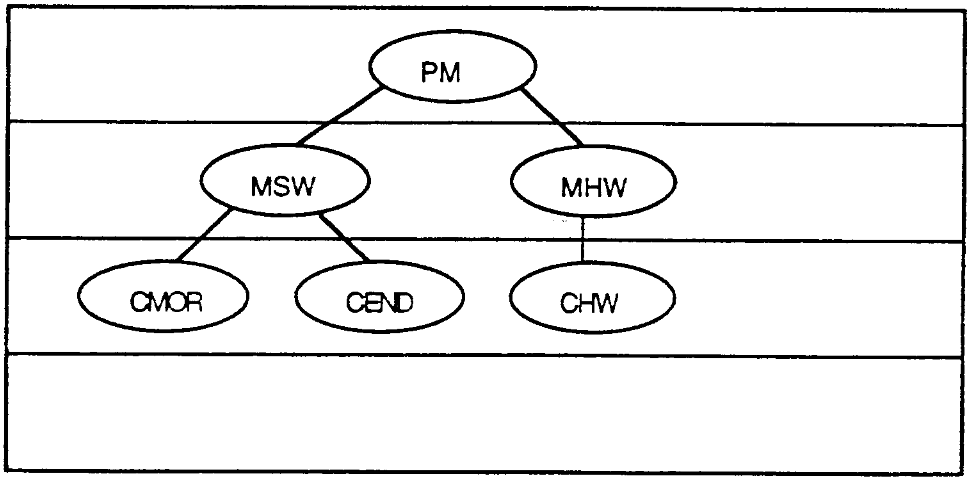


Figure 3a. Blackboard after PM is configured.

ORIGINAL PAGE IS
OF POOR QUALITY

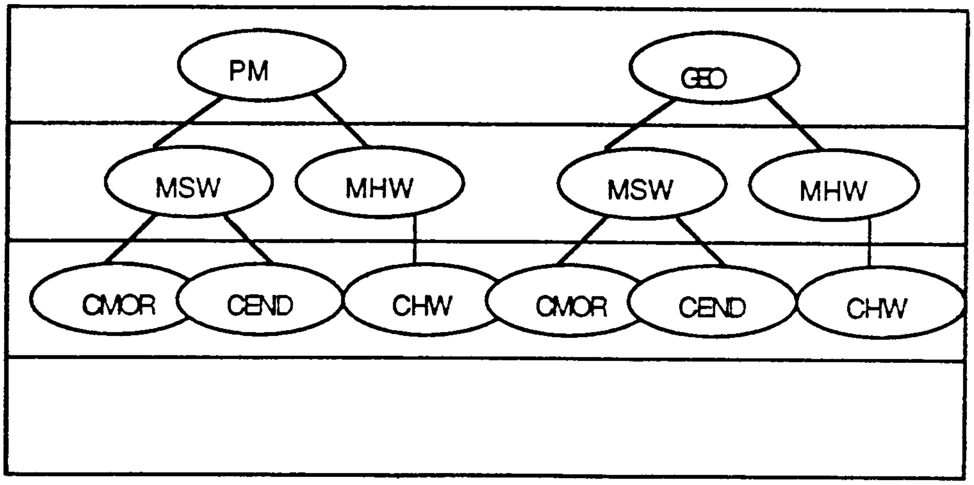
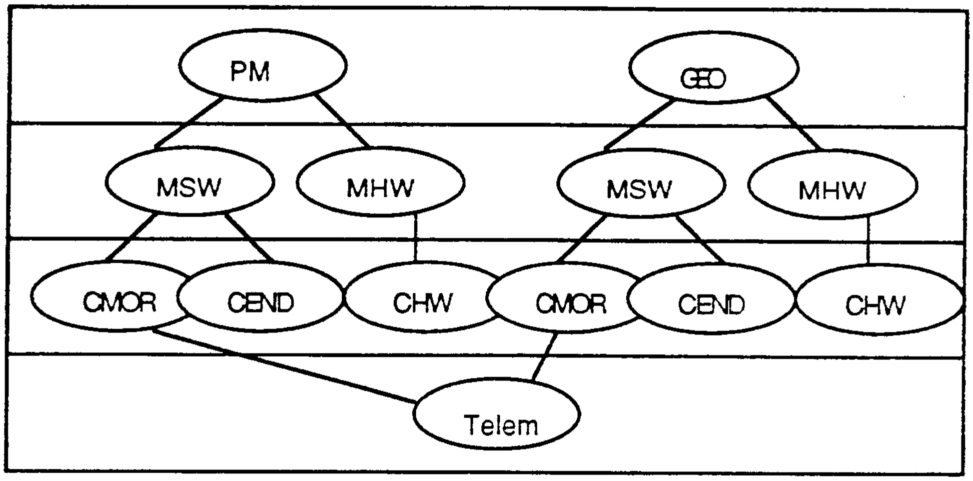


Figure 3b. Blackboard after GEO is configured.



Telem is interpreted as supporting CMOR for PM, CMOR for GEO

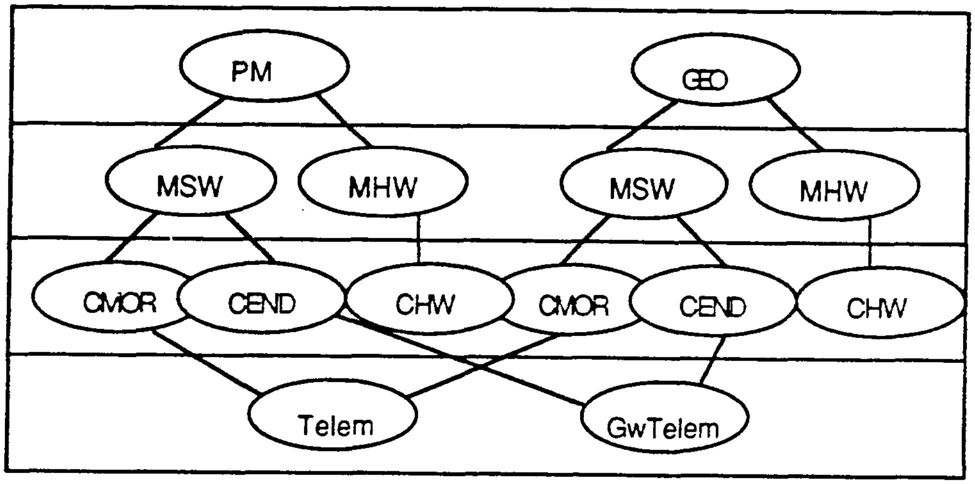
Figure 3c. Blackboard after Telem page request.

support the tasks of checking the endpoint equipment (CEND) of current missions. Upon examining the tasks level of the blackboard, the knowledge sources find that two eligible tasks are posted: the CEND tasks for PM and GEO. Thus, the "GwTelem" action is posted and connected to the CEND tasks. The resulting blackboard is shown in Figure 3d.

5). One of the components used by PM experiences a hardware failure. The component in this example is RUP2. Upon the occurrence of this triggering event, ACTIN's model-driven knowledge sources post a plan to replace the failed component, along with the four associated tasks of finding a currently available replacement (FCUR), finding the duration of the mission (FDUR), finding an unscheduled replacement (FUSC), and executing the replace command (XRPL). The resulting blackboard is shown in Figure 3e.

6). The operator again requests the main telemetry page. This time ACTIN's knowledge sources determine that this action can support three tasks on the blackboard: FDUR for RUP2 and CMOR for both PM and GEO. The resulting blackboard is shown in Figure 3f.

7). The operator requests the schedule for RUP1 ("Rup1Sched"). ACTIN's data-driven knowledge sources determine that the current action type is "Rup1Sched" and that actions of this type potentially support the task of finding unscheduled equipment (FUSC) for RUP components. Upon examining the tasks



GwTelem is interpreted as supporting CEND for PM, CEND for GEO

Figure 3d. Blackboard after GwTelem page request.

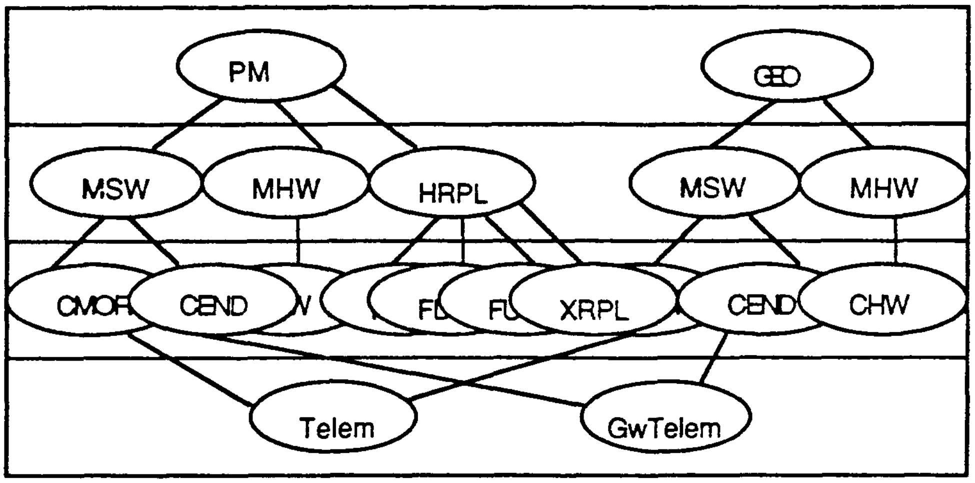
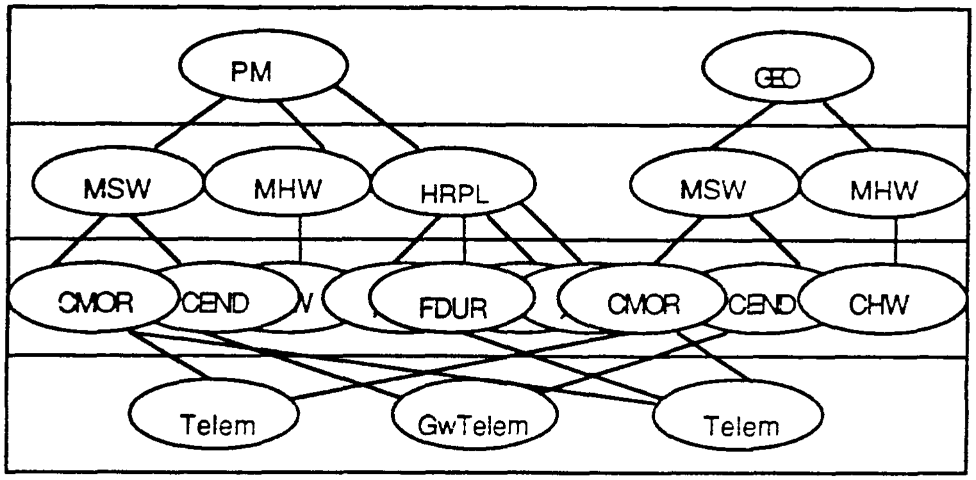


Figure 3e. Blackboard after RUP2 hardware failure.



Telem is interpreted as supporting CMOR for PM, CMOR for GEO, FDUR for RUP2

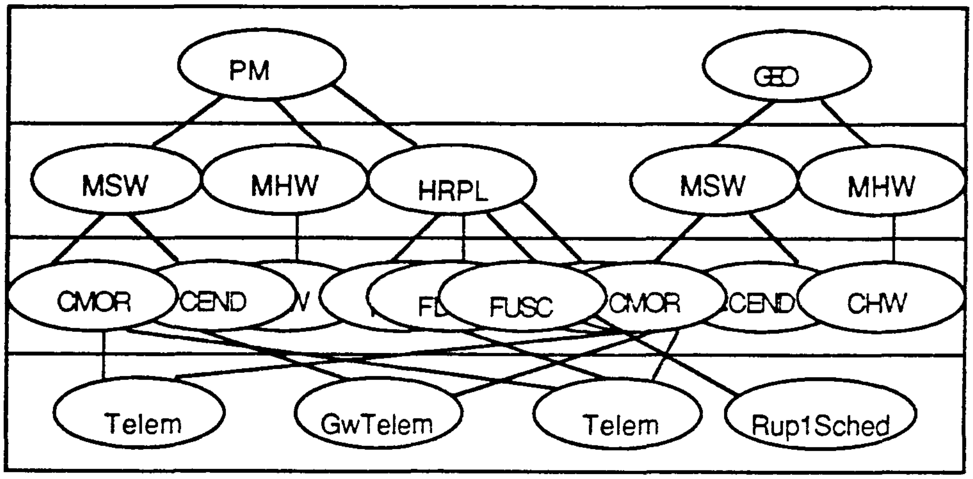
Figure 3f. Blackboard after Telem page request.

level of the blackboard, the knowledge sources find that one eligible task is posted: the FUSC task for RUP2. Thus, the "Rup1Sched" action is posted and connected to the FUSC task associated with the RUP2 replace plan. The resulting blackboard is shown in Figure 3g.

8). Finally, the operator requests the schedule for NAS5. ACTIN's data-driven knowledge sources determine that this request potentially supports finding unscheduled NAS components (i.e., the FUSC task associated with any NAS component). However, although a FUSC type task is posted, it is not associated with a NAS type component. ACTIN is unable to interpret this request as supporting any current tasks. Thus, the "Nas5Sched" request action is posted, but not connected to any current tasks. Figure 3h illustrates the resulting blackboard.

Several characteristics of ACTIN's interpretation algorithm are notable. First, actions are immediately connected to whatever appropriate tasks exist on the blackboard *at the time the actions are posted.* Connection links are not inferred after the action is posted.

Another important feature is ACTIN's property of maximal connectivity. That is, ACTIN interprets actions in the broadest possible context, assuming that the operator is extracting the maximum amount of information from the display pages requested. In the example above, ACTIN inferred that the second telem



Rup1Sched is interpreted as supporting FUSC for RUP2

Figure 3g. Blackboard after Rup1 Schedule request.

ORIGINAL PAGE IS
OF POOR QUALITY

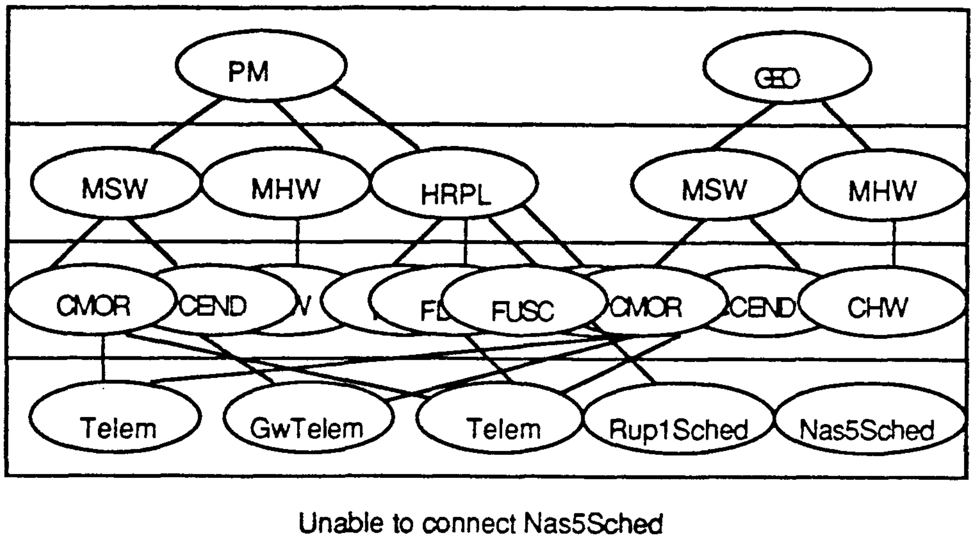


Figure 3h. Blackboard after Nas5 Schedule Request.

action supported all current CMOR tasks as well as the FDUR task for RUP2. Thus, the operator is "given the benefit of the doubt" in the evaluation of performance.

The evaluation of operator performance is performed by knowledge sources that assess the degree to which operator actions support current tasks (and by extension, plans and goals). ACTIN schedules assessments periodically in the context of particular goals or plans. In the example above, ACTIN schedules separate assessments for the control of current mission goals for PM and GEO, and the replace plan for RUP2. Assessments note the number of supporting actions and the time at which those actions occurred. The assessments for PM and GEO would note that the CMOR task is supported by two actions and the CEND task is supported by one action. RUP2's replace plan assessment would state that one action supports the FDUR task and one action supports the FUSC task. The results of these assessments are written to a logfile.

To summarize, the proposed model for intent inferencing uses the OFM methodology to postulate operator functions, subfunctions, and tasks on the basis of current system state and observed operator actions. This model has been implemented using a blackboard architecture. This structure, of which the scenario described in this section is an example, defines the context for intent inferencing.

The OFM and its implementation in ACTIN is an example of "the middle ground" in theory construction in cognitive science (Miller, Polson, and Kintsch, 1984). The theory has well-defined structures and processes to "support both the instantiation of the theory as an executable computer program and qualitative experimental studies of the theory" (Miller, Polson, and Kintsch, 1984, p. 13). In the next section the validation of the proposed model is explored. A two-stage framework for validation is proposed, and experimental results are briefly discussed.

EXPERIMENTAL VALIDATION

Validation of intent inferencing assures that the system is correctly inferring the intentions of the human operator. Within the context of the OFM structure of intentions, this means that the system infers support for the same tasks (and by extension, plans and goals) as the human, given the same set of operator actions. The "human" in this case can be a human domain expert performing a post hoc analysis, or the human operator giving an (on-line) account of intentions. Thus, the proposed two-part framework for the validation of intent inferencing is 1.) comparison of expert and OFMspert analyses; and 2.) comparison of concurrent verbal protocols and OFMspert analysis (see Jones, 1988, for more detail).

The experimental validation of ACTIN's intent inferencing was conducted in two studies. In Experiment 1, a domain expert's interpretations of operator data were compared to ACTIN's interpretations of those same actions on an action-by-action basis. In Experiment 2, verbal protocols were collected from GT-MSOCC operators while they were controlling GT-MSOCC. Statements of intentions for each action were compared to ACTIN's interpretations.

The results of these studies are discussed in detail in Jones (1988). Overall, the results showed that ACTIN's intent inferencing ability compared favorably to inferences made by a domain expert and statements from verbal reports.

ORIGINAL PAGE IS
OF POOR QUALITY

ACKNOWLEDGEMENTS

This research was supported by NASA Goddard Space Flight Center Contract Number NAS5-28575 (Karen Moe and Walt Truszkowski, technical monitors) and by NASA Ames Research Center Grant Number NAG-413 (Dr. Everett Palmer, technical monitor), awarded to Dr. Christine M. Mitchell.

REFERENCES

Chambers, A. B. and Nagel, D. C., 1985, Pilots of the future: Human or computer? *Communications of the ACM, 28,* 11, 1187-1199.

Cohen, A. and Feigenbaum, E. A., 1982, *The Handbook of Artificial Intelligence.* Reading, MA: Addison-Wesley.

Jones, P. M., 1988, Constructing and validating a model-based operator's associate for supervisory control. Report No. 88-1, Center for Human-Machine Systems Research, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

Jones, P. M. and Mitchell, C. M., 1987, Operator modeling: Conceptual and methodological distinctions. *Proceedings of the 31st Annual Meeting of the Human Factors Society, 1,* 31-35. Santa Monica, CA: Human Factors Society.

Miller, J. R., Polson, P. G., and Kintsch, W., 1984, Problems of methodology in cognitive science. In W. Kintsch, J. R. Miller, and P. G. Polson (Eds.), *Method and tactics in cognitive science.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Miller, R. A., 1985, A systems approach to modeling discrete control performance. In W. B. Rouse, (Ed.), *Advances in Man-Machine Systems Research,* Vol. 2, 177-248. New York: JAI Press.

Mitchell, C. M., 1987, GT-MSOCC: a research domain for modeling human-computer interaction and aiding decision making in supervisory control systems. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-17,* 553-572.

Mitchell, C. M. and Forren, M. G., 1987, Multimodal user input to supervisory control systems: Voice-augmented keyboards. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-17,* 594-607.

Mitchell, C. M. and Miller, R. A., 1986, A discrete control model of operator function: A methodology for information display design. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-16,* 343-357.

Mitchell, C. M. and Saisi, D. L., 1987, Use of model-based qualitative icons and adaptive windows in workstations for supervisory control systems. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-17,* 573-593.

Nii, H. P., Feigenbaum, E. A., Anton, J. J., and Rockmore, A. J., 1982, Signal-to-symbol transformation: HASP/SIAP case study. Heuristic Programming Project, Report No. HPP-82-6. Heuristic Programming Project, Stanford University, Stanford, CA.

Nii, H. P., 1986, Blackboard systems. *AI Magazine, 7-2, 7-3.*

Rasmussen, J., 1986, *Information processing and human-machine interaction: An approach to cognitive engineering.* New York: North-Holland.

Rouse, W. B., Geddes, N. D., and Curry, R. E., 1987, An architecture for intelligent interfaces: Outline of an approach to supporting operators of complex systems. *Human-Computer Interaction, 3.*

Rubin, K. S., Jones, P. M. and Mitchell, C. M., 1987, OFMspert: Application of a blackboard architecture to infer operator intentions in real time decision making. Report No. 87-6, Center for Human-Machine Systems Research, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA. Also *IEEE Transactions on Systems, Man, and Cybernetics,* to appear.

Schank, R. C. and Abelson, R. P., 1977, *Scripts plans goals and understanding.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Sheridan, T. B. and Johannsen, G., 1976, *Monitoring Behavior and Supervisory Control.* New York: Plenum.

Wenger, E., 1987, *Artificial intelligence and tutoring systems.* Los Altos, CA: Morgan Kaufmann.

Wickens, C. D., 1984, *Engineering Psychology and Human Performance.* Columbus, OH: Charles Merrill.

Winograd, T., 1972, *Understanding natural language.* New York: Academic Press.

Woods, D. D., 1986, Cognitive technologies: The design of joint human-machine cognitive systems. *The AI Magazine,* 86-92.